previsao_nivel_rio

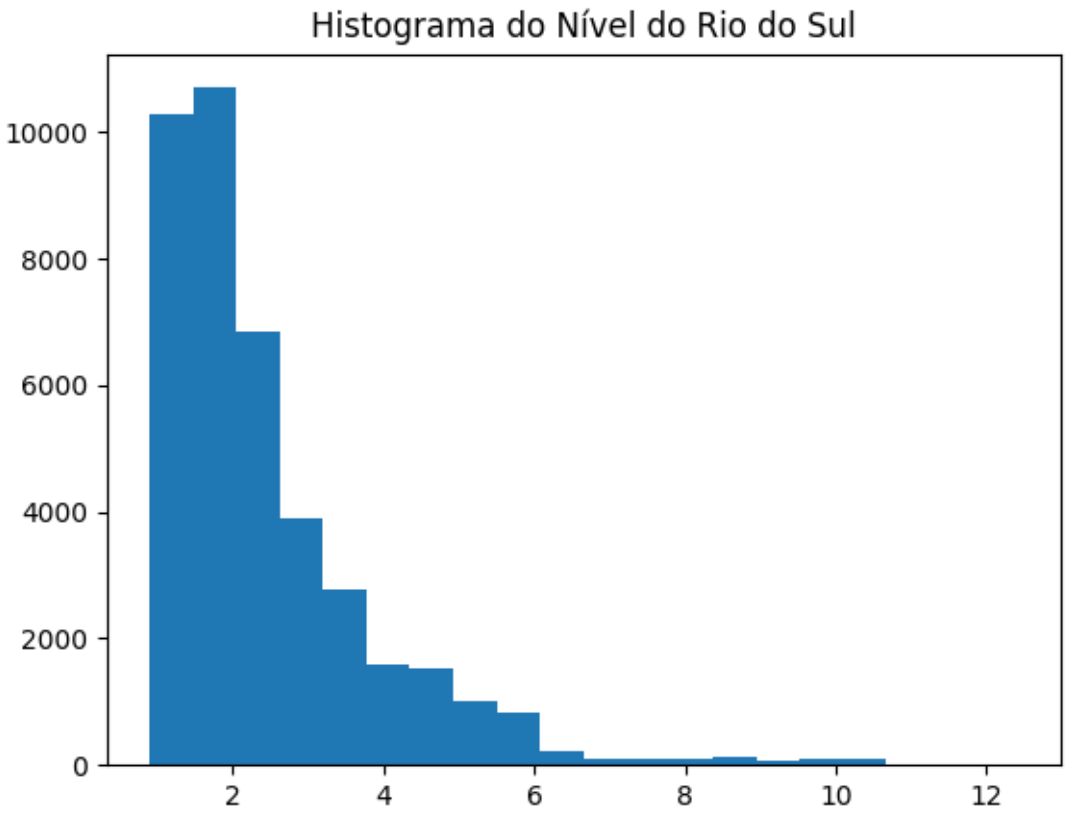
July 19, 2025

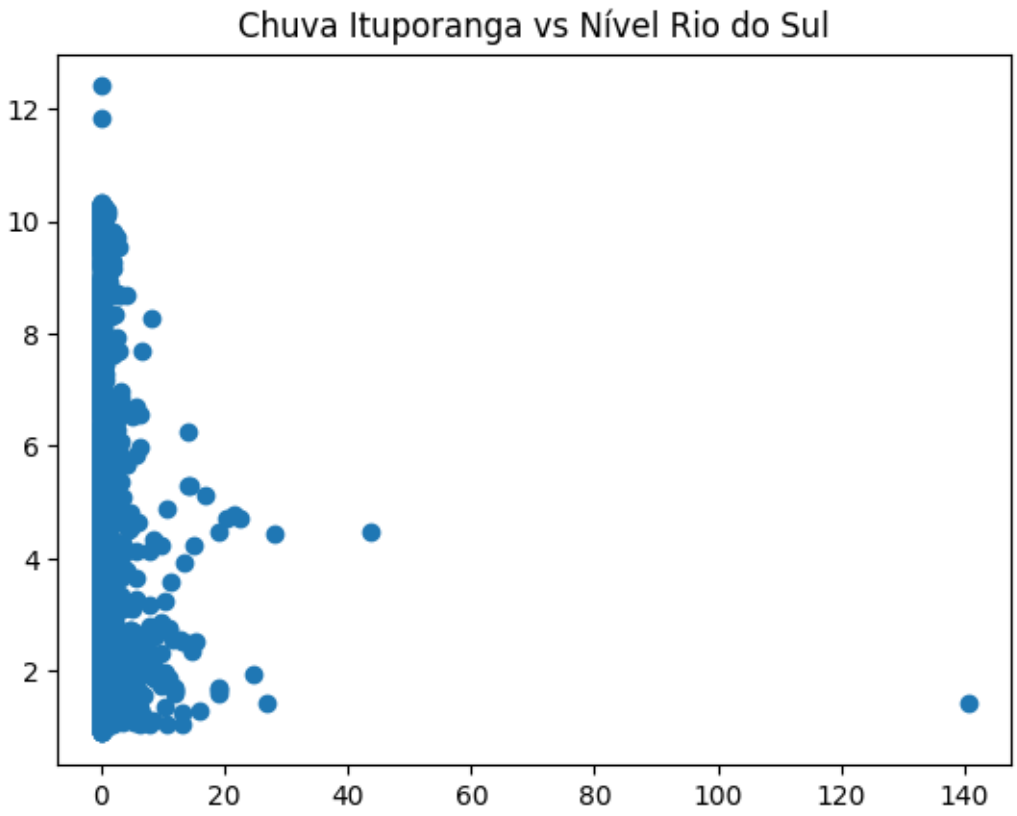
[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import joblib
from sklearn.linear_model import LinearRegression
from sklearn.metrics import root_mean_squared_error, mean_absolute_error,␣
↪r2_score
```

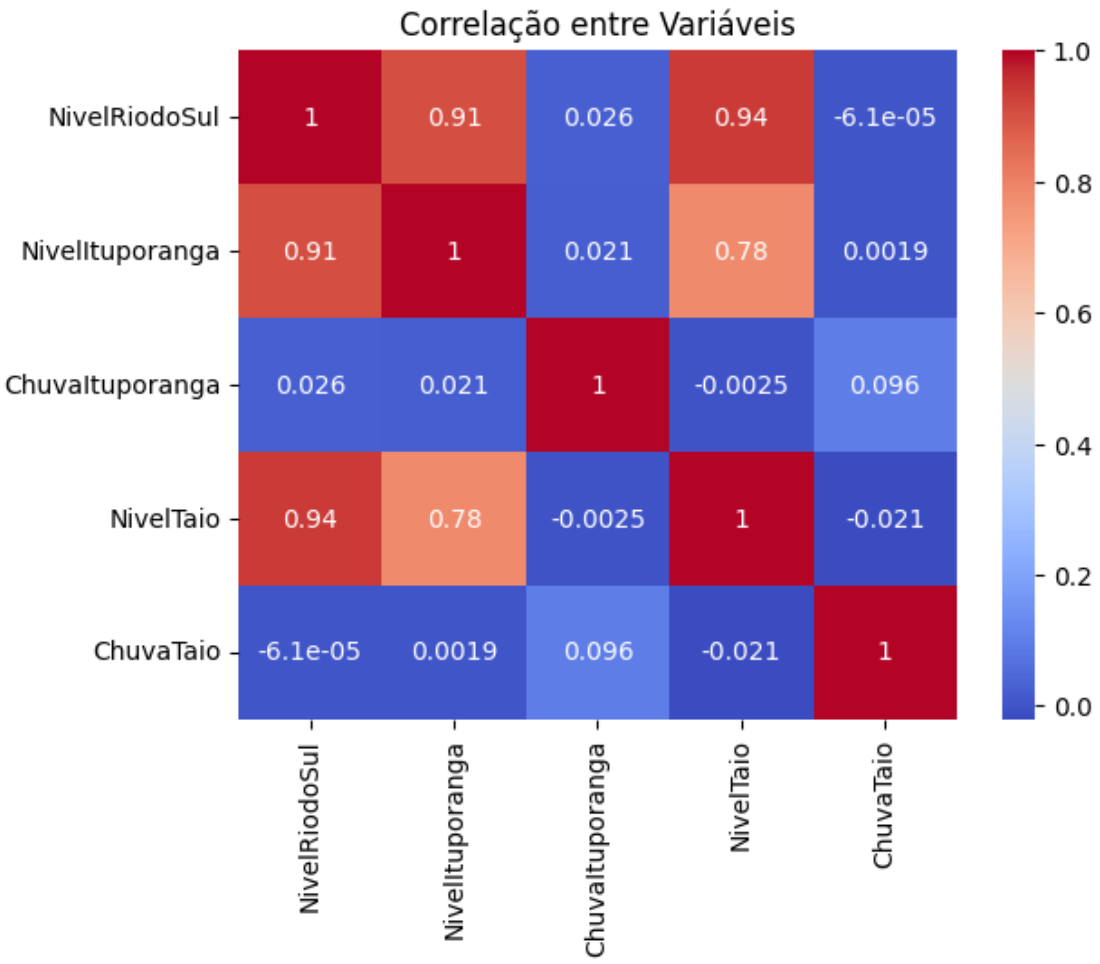
[2]:
```
# 1. Leitura e Análise Inicial
df = pd.read_excel('dados_rio_e_chuva.xlsx')

# Convert river levels from centimeters to meters
df['NivelRiodoSul'] = df['NivelRiodoSul'] / 100
df['NivelItuporanga'] = df['NivelItuporanga'] / 100
df['NivelTaio'] = df['NivelTaio'] / 100

print(df.isnull().sum())  # Verificar valores faltantes
df = df.dropna()  # Remover linhas com valores faltantes
print(df.duplicated().sum())  # Verificar duplicatas
df = df.drop_duplicates()  # Remover duplicatas

# Histograma da variável alvo
plt.hist(df['NivelRiodoSul'], bins=20)
plt.title('Histograma do Nível do Rio do Sul')
plt.show()

# Scatter plot
plt.scatter(df['ChuvaItuporanga'], df['NivelRiodoSul'])
plt.title('Chuva Ituporanga vs Nível Rio do Sul')
plt.show()

# Heatmap de correlação
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```

```
plt.title('Correlação entre Variáveis')
plt.show()
```

```
NivelRiodoSul       0
NivelItuporanga     0
ChuvaItuporanga     0
NivelTaio           0
ChuvaTaio           0
dtype: int64
25131
```




Chuva Ituporanga vs Nível Rio do Sul
12
10
8
6
4
2
0
20
40
60
80
100
120
140



[3]:
```
# 2. Pré-processamento
X = df[['NivelItuporanga', 'ChuvaItuporanga', 'NivelTaio', 'ChuvaTaio']]  #␣
 ↪Variáveis de entrada
y = df['NivelRiodoSul']  # Variável alvo
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
joblib.dump(scaler, 'scaler.pkl')  # Save the fitted scaler
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3,␣
 ↪random_state=42)
```

[4]:
```
# 3. Treinamento do Modelo
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# Avaliação
rmse = root_mean_squared_error(y_test, y_pred)
```

```
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f'RMSE: {rmse}, MAE: {mae}, R²: {r2}')
```

```
RMSE: 0.2927148150262595, MAE: 0.18548993125118815, R²: 0.9572045748376021
```

[5]:
```
# 4. Salvar o Modelo
joblib.dump(model, 'modelo_previsao_rio.pkl')
```

[5]:
```
['modelo_previsao_rio.pkl']
```

[6]:
```
print("Autor: Victor Miguel A M Falcão")
print("Github: https://github.com/victormiguel22/NivelRio")
!jupyter nbconvert --to pdf previsao_nivel_rio.ipynb
```

```
Autor: Victor Miguel A M Falcão
Github: https://github.com/victormiguel22/NivelRio
[NbConvertApp] Converting notebook previsao_nivel_rio.ipynb to pdf
[NbConvertApp] Support files will be in previsao_nivel_rio_files/
[NbConvertApp] Making directory ./previsao_nivel_rio_files
[NbConvertApp] Writing 52757 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 111846 bytes to previsao_nivel_rio.pdf
```